

DeCREE:  
Eine Domänenspezifische Sprache  
zur Modellierung von  
Choreography-First Szenarien

Philipp Gringel  
Stefan Gudenkauf  
Steffen Kruse

Veröffentlicht in:  
Multikonferenz Wirtschaftsinformatik 2012  
Tagungsband der MKWI 2012  
Hrsg.: Dirk Christian Mattfeld; Susanne Robra-Bissantz



Braunschweig: Institut für Wirtschaftsinformatik, 2012

# DeCREE: Eine Domänenspezifische Sprache zur Modellierung von Choreography-First Szenarien

Philipp Gringel, Stefan Gudenkauf, Steffen Kruse

OFFIS – Institut für Informatik, 26121 Oldenburg,

E-Mail: {philipp.gringel | stefan.gudenkauf | steffen.kruse}@offis.de

## Abstract

Die Choreografie-Diagramme der BPMN 2.0 erlauben die Modellierung von Interaktionsprotokollen zwischen verschiedenen Teilnehmern. Es ist sinnvoll, die jeweiligen Zuständigkeiten der Teilnehmer explizit mit zu modellieren. Herausforderungen sind die strikte Einhaltung der relevanten BPMN-Untermenge zur Erhaltung von Konformität und Interoperabilität, sowie die Handhabung der Zuständigkeiten als Modellelemente erster Klasse um Interaktionsprotokolle auf fachlich höherer Ebene validieren zu können. Mit dem vorgeschlagenen *Choreography-First* Ansatz und der Modellierungssprache *DeCREE* bewältigen wir diese Herausforderungen. Das Metamodell von *DeCREE* umfasst sowohl eine BPMN-Untermenge als auch RACI-Zuständigkeiten. Als Beleg für die Machbarkeit stellen wir eine Implementierung der DSL auf Basis des Xtext Language Frameworks vor.

## 1 Motivation

Das Management von organisationsübergreifenden Geschäftsprozessen macht es notwendig, den Nachrichtenaustausch zwischen den einzelnen Teilnehmern explizit zu handhaben. Choreografien stellen hierzu eine Abstraktion dar, die die Reihenfolge des Nachrichtenaustauschs in den Vordergrund stellt. Eine Choreografie definiert die genaue Reihenfolge dieses Nachrichtenaustauschs unabhängig von den Prozessen der einzelnen Teilnehmer [1]. Die eigentlichen internen Prozesse der Teilnehmer sind dabei nur soweit von Interesse, als dass sie die zu sendenden oder zu empfangenden Nachrichten korrekt und in der korrekten Reihenfolge implementieren. Mit dem Erscheinen der Business Process Model and Notation (BPMN) in der Version 2.0 ist es nun möglich, Choreografien formal zu modellieren und die Schnittstellen zum Entwurf der organisationsinternen Prozesse abzuleiten [5, S. 325].

Werden Choreografien frühzeitig im Entwicklungsprozess organisations-übergreifender Geschäftsprozesse modelliert, können sie als *vertragliches Kommunikationsprotokoll* zwischen den beteiligten Teilnehmern angesehen und als Rahmen für den Prozessentwurf genutzt werden. Dies spiegelt sich auch in der Definition von Choreografien in der BPMN 2.0 Spezifikation wieder: „A Choreography is a definition of expected behavior, basically a

procedural business contract, between interacting Participants” [5, S. 315]. Die vertragliche Vereinbarung umfasst dabei nicht nur die Spezifikation der durchzuführenden kollaborativen Aktivitäten, sondern legt auch die Beteiligung der jeweiligen Teilnehmer fest. Außerdem können durch das frühzeitige Modellieren von Choreografien Inkompatibilitäten der Prozessentwürfe der einzelnen Teilnehmer und ein zusätzlicher Aufwand durch später durchzuführende Restrukturierungen vermieden werden.

In Organisationen gibt es eine Vielzahl von Rollen, die von unterschiedlichen Personen ausgefüllt werden können. Dabei können folgende, grundlegende Überlegungen angestellt werden [9]: Das *Rollenverständnis*, das eine Person seiner Rolle entgegen bringt, kann von den *Erwartungen* Dritter an die Rolle abweichen. Diese Erwartungen werden in der Regel von den erwarteten Arbeitsergebnissen der Rolle definiert. Davon kann wiederum das eigentliche *Rollenverhalten* der die Rolle ausfüllenden Person abweichen. Dem dadurch entstehenden Spannungsverhältnis in kollaborativen Geschäftsprozessen kann mit einem Abgleich der Rollenverständnisse aller Beteiligten begegnet werden. Daher ist es bei der vertraglichen Vereinbarung der Interaktionsprotokolle notwendig, zusätzlich zur Spezifikation der kollaborativen Aktivitäten und den jeweiligen Beteiligungen auch das Rollenverständnis der einzelnen Teilnehmer in einer Choreografie vertraglich zu vereinbaren. Hierzu kann die von Smith und Erwin beschriebene RACI-Methode verwendet werden [9], um durch die explizite Beschreibung von *Zuständigkeiten* eine Rolle in Beziehung zu einer von ihr durchzuführenden Aktivität zu setzen. Als Folge können aus unterschiedlichen Rollenverständnissen entstehende Zuständigkeitskonflikte in Choreografien frühzeitig aufgedeckt und vermieden werden.

Der in diesem Beitrag vorgestellte Ansatz *Choreography-First* adressiert die formale Modellierung von Choreografien in einer frühen Phase des Entwicklungsprozess und umfasst die Modellierung des Interaktionsprotokolls zwischen Teilnehmern und deren gemeinsames Rollenverständnis. Hierzu werden die BPMN 2.0-Sprachelemente zur Modellierung von Choreografie-Diagrammen um Elemente für RACI-Zuständigkeiten erweitert. Damit können sowohl das Interaktionsprotokoll als auch die Zuständigkeiten der Teilnehmer für den späteren Prozessentwurf frühzeitig vertraglich vereinbart werden. Für die technische Umsetzung betrachten wir die folgenden Anforderungen (Tabelle 1):

Schlüssel	Anforderung
A1	BPMN 2.0-Konformität: Als de facto-Standard genießt die BPMN 2.0 eine hohe Verbreitung. Sie unterstützt Choreografien nativ.
A2	Interoperabilität: Die Umsetzung soll interoperabel zu gängigen BPMN 2.0-Werkzeugen sein um Medienbrüche im Entwicklungsprozess so gering wie möglich zu halten.
A3	Iteratives Vorgehen: Die Umsetzung soll iterative Vorgehen bestmöglich unterstützen. Modellanpassungen in späteren Phasen des Entwicklungsprozesses sollen ein Minimum an weiteren Änderungen verursachen.
A4	Usability: Die Umsetzung soll aus Nutzersicht möglichst komfortabel sein und gängige Erwartungen an modernen Werkzeugen gerecht werden.
A5	Validierung: Die Umsetzung soll die strukturelle und an der Fachdomäne orientierte Validierung der Modelle ermöglichen und frühzeitig auf Fehler hinweisen. Wenn möglich sollen standardisierte Lösungen bereitgestellt und automatisch durchgeführt werden.

**Tabelle 1: Anforderungen an die technische Umsetzung von Choreography-First**

Mit über 150 Entitäten im Metamodell und mindestens 11 Diagrammtypen auf verschiedenen Abstraktionsebenen ist die BPMN 2.0-Spezifikation komplex [5, S. 41]. Die Beschränkung auf die wesentlichen Aspekte der betrachteten Anwendungsdomäne fällt auch erfahrenen Fachmodellierern schwer. Die gängige Praxis empfiehlt daher, zunächst eine Untermenge der BPMN zu identifizieren und anschließend ausschließlich diese zu verwenden [10]. Die strikte Einhaltung dieser Untermenge liegt dabei im Verantwortungsbereich der Fachmodellierer. Reine BPMN 2.0 Modellierungswerkzeuge können diese Konformität üblicherweise nicht erzwingen. Zusätzlich macht die Modellierung von RACI-Zuständigkeiten eine Erweiterung notwendig, die in der BPMN 2.0 selbst nicht vorgesehen ist. Grundsätzlich sieht die Spezifikation einen Erweiterungsmechanismus über so genannte Extensions vor [5, S. 57ff.]. Dieser hat jedoch schwerwiegende Nachteile: Erstens ist die fachliche Validierung der Erweiterungen nicht vorgesehen (**A5**). BPMN-Werkzeuge setzen zwar die strukturelle Konformität von Modellen zum BPMN 2.0-Metamodell durch, erlauben aber keine fachliche Prüfung von Extensions. Zweitens ist es fraglich, ob über Werkzeugketten hinweg neue Notationselemente einheitlich handgehabt und ausgetauscht werden können (**A4**). Drittens können Extensions die Interoperabilität von Werkzeugen brechen, da BPMN 2.0 ihre Weitergabe in Werkzeugketten nicht garantiert (**A3** und **A4**) [5, S. 57].

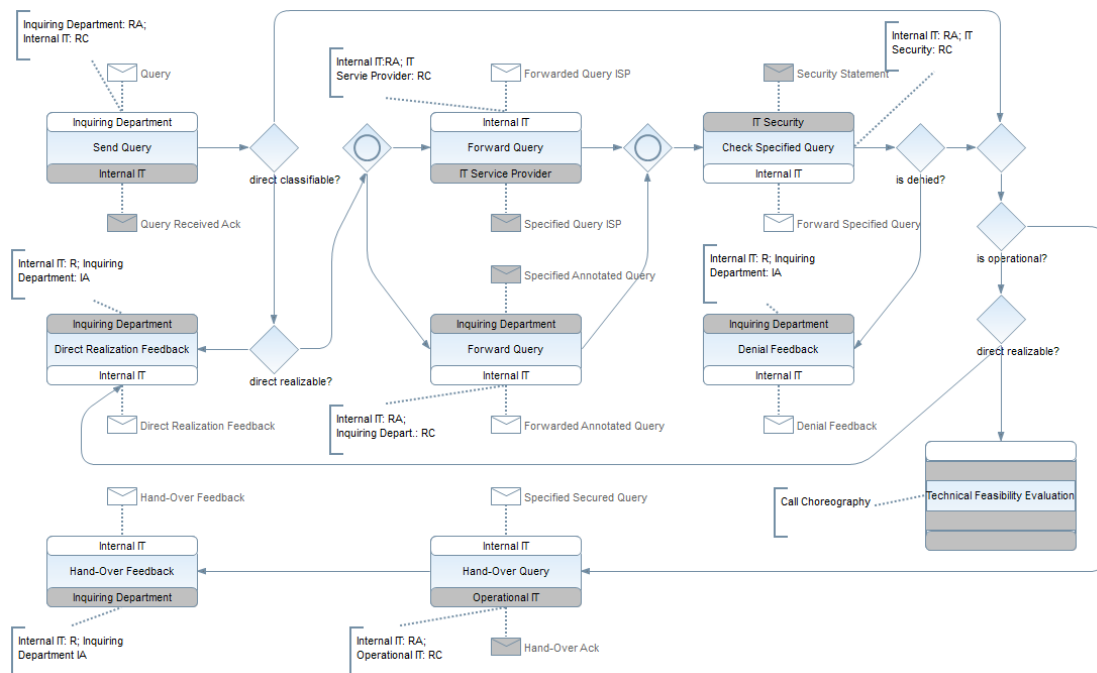
In der Konsequenz wurde der folgende technische Ansatz gewählt: Im ersten Schritt wurde eine Untermenge des BPMN Metamodells gewählt, dass genau die Elemente enthält, die für die minimale Modellierung von Choreografien nötig sind (Anforderung **A1**). Diese Untermenge wurde dann um Elemente für die RACI-Zuständigkeiten erweitert und als Metamodell formalisiert. Die durch das Metamodell repräsentierte abstrakte Syntax wurde anschließend um eine konkrete Syntax ergänzt, um die textuelle domänenspezifische Sprache *DeCREE* für um Zuständigkeiten erweiterte Choreografien zu erhalten (DSL for Choreography Responsibility Engineering). Die Semantik der Sprache wird von der BPMN 2.0 Spezifikation und der RACI-Methode bereitgestellt. Mit dem Eclipse Xtext Framework [11] wurden ein Parser für die Sprache und eine dazugehörige Werkzeugumgebung als Eclipse-Plugin entwickelt und um Mechanismen zur Validierung wie Warnungen bei Validierungsfehlern und automatischen Quickfixes erweitert (Anforderungen **A4** und **A5**). Die so entstandene Arbeitsumgebung stellt eine konsistente, auf den Fachexperten ausgerichtete Werkzeugunterstützung für um Zuständigkeiten erweiterte Choreografien dar.

Die separate Arbeitsumgebung erfordert Mechanismen zur Interoperabilität zwischen *DeCREE* und BPMN. Hierzu wurde eine auf dem Query/View/Transformation-Standard (QVT) basierende bidirektionale Transformation entwickelt, deren Regeln den Zusammenhang zwischen Elementen von *DeCREE* und der BPMN beschreiben und die automatisierte Transformation konformer Elemente zwischen den beiden Sprachen erlauben. Dabei wurden die für RACI hinzugefügten Elemente auf BPMN-Annotationen abgebildet, damit diese von standardkonformen BPMN 2.0-Werkzeugen gelesen werden können (Anforderung **A2**). Nach der erstmaligen Transformation von DSL zu BPMN 2.0 werden die erzeugten BPMN 2.0-Modelle um einfache Layout-Informationen angereichert, um diese auch grafisch darstellbar zu machen. Zusätzlich bietet die QVT-Transformation die Möglichkeit, bei der erneuten Transformation von Modellen auf die Trace-Informationen vorhergehender Ausführungen zurückzugreifen und so Updates von Modellen nach Veränderungen von Modellteilen durchzuführen. Dies ermöglicht iterative Entwicklungsprozesse: Nachträgliche

Änderungen können mit minimalen Änderungen auf BPMN 2.0-Modelle automatisch übertragen werden, ohne Änderungen auf Seiten der BPMN-Modelle zu überschreiben (Anforderung A3).

## 2 Szenario

Bild 1 beschreibt das anonymisierte Szenario „Fachliche Machbarkeit prüfen“ aus dem IT-Anforderungsmanagement eines IT-Consulting-Unternehmens als BPMN Choreografie. Im Fokus stehen dabei durch Kontrollflüsse geordnete Choreografie-Aktivitäten. Sie repräsentieren den Austausch einer oder mehrerer Nachrichten zwischen zwei oder mehreren Teilnehmern. Der initiiierende Teilnehmer wird dabei als weißes Band, die reagierenden als graues Band an die Choreografie-Aktivität annotiert. Analog wird die initiiierende Nachricht weiß, die Rücksendungen grau schattiert dargestellt. Der Kontrollfluss zwischen den einzelnen Aktivitäten erfolgt dabei analog zu herkömmlichen BPMN Prozessmodellierungen (Sequenzflüsse, Entscheidungen, Ereignisse), ist aber hinsichtlich der Anzahl der zugelassenen Elemente eingeschränkt.



**Bild 1:** Szenario „Fachliche Machbarkeit prüfen“ aus dem IT-Auftragsmanagement

Gegenstand des Szenarios ist die Prüfung einer Anforderung, die eine Fachabteilung eines Unternehmens an die unternehmensinterne IT-Abteilung (Interne IT) stellt. Dies kann beispielsweise eine einfache Porttfreigabe sein. Die Anforderung wird durch die Interne IT zunächst auf ihre fachliche Machbarkeit geprüft und bei direkter Realisierbarkeit sofort umgesetzt. Weitere Beteiligte sind dabei die IT-Sicherheitsabteilung sowie gegebenenfalls ein externer IT Service Provider, von dem gegebenenfalls zusätzliche Informationen eingeholt werden müssen. Sollte die Umsetzung der Anfrage in den Aufgabenbereich der Operationalen IT fallen, wird die Anfrage an die Operationale IT übergeben. Alternativ ist die weitere technische Machbarkeitsprüfung notwendig.

Wie das Szenario zeigt, sind die Zuständigkeiten der einzelnen Teilnehmer an den Choreografieaktivitäten mit Hilfe der RACI-Technik annotiert [9]. Dabei werden Zuständigkeiten danach modelliert, ob ein Teilnehmer durchführungsverantwortlich (*responsible*, R), rechnungspflichtig oder kostenverantwortlich (*accountable*, A), zu konsultieren (*consulted*, C), oder zu informieren (*informed*, I) ist. Dies kann als naheliegende Erweiterung des Choreografiemodells angesehen werden, da die BPMN 2.0 Spezifikation die Notwendigkeit von Zuständigkeiten bei Choreografieaktivitäten mit mehreren Teilnehmern erkennt, aber nicht weiter auf deren Spezifikation eingeht: „[I]f there are more than two Participants, then the modeler needs to be careful to sequence the Choreography Activities in such a way that the Participants know when they are responsible for initiating the interactions.“ [S. 317, BPMN]. Mit RACI annotierte Choreografieaktivitäten erlauben es, Unstimmigkeiten in der Choreografiemodellierung leicht zu erkennen. Beispielsweise sollte immer nur ein Teilnehmer pro Choreografieaktivität *accountable* sein und der initiiierende Teilnehmer immer *responsible*. In der BPMN 2.0 sind RACI-Zuständigkeiten jedoch keine Elemente erster Klasse und können nur als Textannotationen oder den bereits genannten Extensions modelliert werden. Eine Validierung auf fachliche Korrektheit der um Zuständigkeiten erweiterten Choreographien ist nicht möglich.

### 3 Verwandte Arbeiten

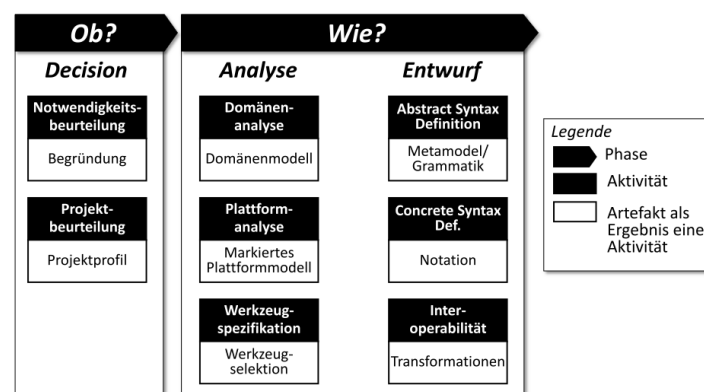
Domänenspezifische Sprachen (DSLs) sind Sprachen, die auf eine bestimmte Anwendungs- oder Fachdomäne ausgerichtet sind und ihre Allgemeingültigkeit gegen eine erhöhte Ausdruckskraft in der entsprechenden Domäne tauschen [3]. Sie zielen auf eine Verbesserung der Produktivität in der Softwareentwicklung ab, indem sie von den technischen Details der verwendeten Plattform abstrahieren. Obwohl die Idee der domänenspezifischen Modellierung, also der abstrahierenden Darstellung von Sachverhalten einer fachlichen Domäne auf möglichst hohem Niveau, schon lange existiert, hat sich ihr Potenzial erst im Rahmen der modellgetriebenen Softwareentwicklung (MDSD) ausgeweitet: Zum einen beschränkt sich die Nutzung von DSLs nicht mehr allein auf die anwendungsnahe Programmierung, sondern betrachtet auch direkt für Fachexperten verständliche Fachmodelle. Zum anderen existieren ausgereifte Frameworks, um die Infrastruktur einer DSL aus der Sprachdefinition größtenteils generieren zu können. Als Beispiel sei hier das eingangs erwähnte Xtext Language Framework genannt. Als Folge dieser Entwicklung können DSLs heutzutage als Standardwerkzeug im Repertoire der Softwareentwicklung aufgefasst werden [12].

Mernik et al. beschreiben in ihrem Artikel Muster, die sie für den Entwurf, die Analyse und die eigentliche **Entwicklung einer DSL** identifiziert haben [3]. Dazu gehören Entwurfsmuster zur Ausnutzung einer bestehenden Wirtssprache. Wird eine bestehende Sprache teilweise genutzt und um fachspezifische Elemente erweitert, bezeichnen sie dies als *Piggybacking*. Da die in diesem Beitrag beschriebene DSL DeCREE die BPMN 2.0 nicht nur teilweise nutzt, sondern auch um neue Konzepte erweitert, kann der vorliegende Beitrag als *BPMN 2.0-Piggybacking* bezeichnet werden.

Zu Beginn der 1990er Jahre wurden die ereignisgesteuerten Prozessketten (EPK) zur **Beschreibung von Geschäftsprozessen** eingeführt. Diese lassen sich durch beschreibende Komponenten erweitern (eEPK) [4]. Diese Komponenten können zum

Beispiel einen Datenfluss oder eine Organisationseinheit beschreiben. Zwischen den Funktionen einer eEPK und einer Organisationseinheit kann beispielsweise eine Relation „führt aus“ genutzt werden, um die Ausführungsverantwortlichkeit der Organisationseinheit für eine Funktion zu beschreiben. Dies kann als einfache Alternative zu RACI-Zuständigkeiten angesehen werden, ist aber beschränkt auf eEPK. Die bereits in den ersten Versionen der BPMN vorhandenen Prozessdiagramme werden zum Beispiel in [13-15] um die Möglichkeit erweitert, **Autorisierungsbeschränkungen auf Task-Ebene** einzuführen. In [13] wird der Schwerpunkt auf Autorisierungsanforderungen gelegt und die BPMN-Modellelemente zu diesem Zweck mit zusätzlichen Eigenschaften angereichert. Der Ansatz verwendet als Grundlage für die Semantikbeschreibung gefärbte Petri Netze und beschränkt sich auf Autorisierungsanforderungen innerhalb eines Prozesses, also auf Ebene der Prozessdiagramme. In [2] wird beschrieben, wie die Prozessdiagramme der BPMN 1.1 um Beschränkungen bzgl. der Ressourcenverteilung erweitert werden können. Die Autoren **erweitern ein Metamodell der BPMN** um entsprechende Klassen, wie zum Beispiel um Rollen, die über eine bestimmte Ressource (z.B. Mitarbeiter, der die Rolle ausfüllt) mit Instanzen von Tasks in Relation stehen. Die Einschränkungen der Zuweisung von Ressourcen zu Tasks werden in OCL (Object Constraint Language) notiert. Um diese Notation vor dem Endnutzer zu verbergen, setzen die Autoren auf ein auf Oryx basierendes Werkzeug in dem die Einschränkungen der Ressourcenverteilung mit Hilfe eines Wizards definiert werden können. Die Rollenzuweisung und die Einschränkungen finden im Werkzeug nach der Definition eines Tasks statt. Rodríguez et al. haben ein **eigenes Metamodell** entwickelt, das sie aus den Prozessdiagrammen der BPMN 1.0 abgeleitet und um Aspekte der Sicherheitsmodellierung erweitert haben [7].

Die beschriebenen Ansätze haben gemein, dass sie sich mit den Kollaborationsdiagrammen der BPMN vor der Version 2.0 befassen. Auch sahen sich die Autoren gezwungen, durch ein vor BPMN 2.0 fehlendes, formales Metamodell, ein eigenes Metamodell zu definieren und dieses gemäß ihren Anforderungen zu erweitern. Es bleibt zu prüfen in wie fern die beschriebenen Ansätze nach der Standardisierung des BPMN Metamodells durch die OMG in 2011 weiterhin ohne Anpassungen genutzt werden können. Eine **BPMN 2.0 Metamodellerweiterung** findet sich in [8]. Dort wird der Erweiterungsmechanismus der BPMN 2.0 (Extensions) genutzt, um sog. *Compliance Scopes* in Prozessdiagrammen definieren zu können. Wie eingangs erwähnt wird dieser Ansatz jedoch nur durch wenige Werkzeuge unterstützt und es fehlen grafische Notationen und Möglichkeiten zur Validierung der fachlich erweiterten Modelle.



**Bild 2:** Vorgehen zur DSL-Entwicklung

## 4 Entwicklung von DeCREE

Zum Entwurf von DeCREE und der Infrastruktur wurde ein auf dem Phasenmodell von Mernik et al. basierendes [3] Vorgehen verwendet. Bild 2 zeigt einen Überblick über die durchgeführten Aktivitäten und deren Ergebnisse. Die Entwicklungsaktivitäten wurden dabei bedarfsgerecht und in kurzen Feedback-Zyklen über die verschiedenen Versionen der Ergebnisse durchgeführt. Zur Beantwortung der Frage, ob eine DSL entwickelt werden soll, sehen wir die Notwendigkeitsbewertung durch Abschnitt 1 als positiv beantwortet an.

### 4.1 Analyse

Zur Beantwortung der Frage, *wie* die DSL entwickelt werden soll, wurde zunächst eine Domänenanalyse durchgeführt. Die Zieldomäne wird als *um RACI erweiterte BPMN 2.0-interoperable, fachlich höherwertige Geschäftschoreografien* bezeichnet. Die anschließende Plattformanalyse lieferte eine Übersicht der relevanten BPMN 2.0 Modellfragmente. Hierzu wurde die für höherwertige Choreografien notwendige minimale Untermenge der BPMN 2.0 identifiziert und die zugehörigen Elemente im Metamodell der BPMN 2.0 ausfindig gemacht (Anforderung **A1**). Die so identifizierten Metamodellfragmente stellten das Ausgangsmaterial für den Entwurf der abstrakten Syntax der DSL sowie für die Modelltransformationen zur BPMN 2.0-Interoperabilität dar (Anforderung **A2**). Die nachfolgende Tabelle 2 zeigt eine Übersicht der identifizierten Elemente zusammen mit einer Beschreibung für die Auswahl. Die fett markierten BPMN-Elemente stellen die primären Konzepte der Fachdomäne dar.

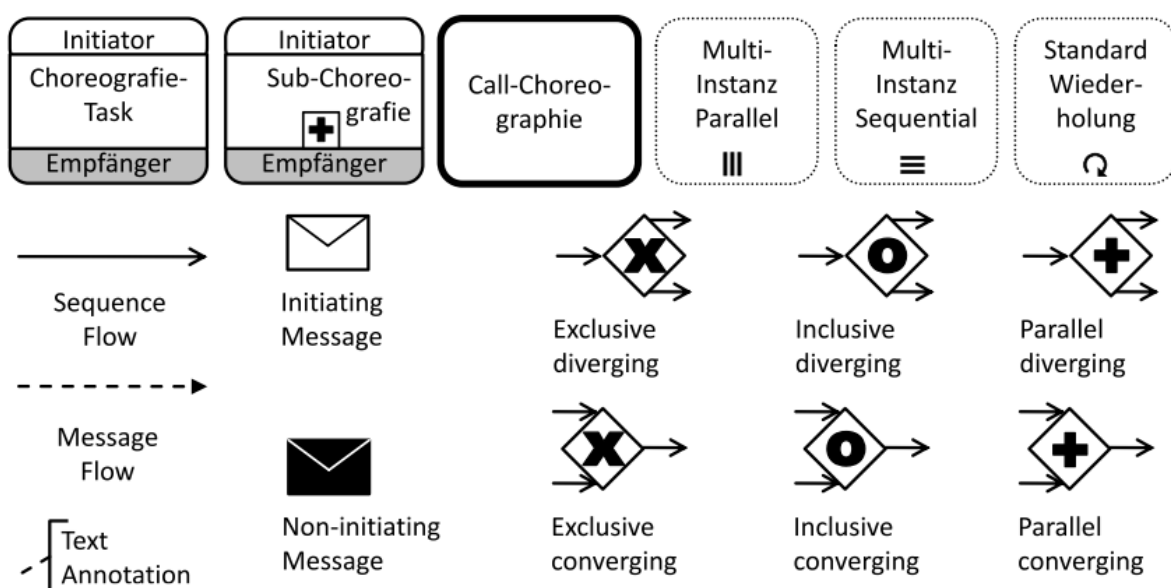
<b>BPMN 2.0 Element</b>	<b>Beschreibung</b>
<b>Choreography</b>	Oberstes Modellelement zur Repräsentation des Konzepts Choreografie
<b>Participant</b>	Repräsentation eines Teilnehmers in organisationsübergreifenden Geschäftsprozessen
<i>FlowElementsContainer</i>	Abstrakte Oberklasse zur Definition der Obermenge aller relevanten Modellelemente in Choreografien
<i>FlowElement</i>	Abstrakte Oberklasse aller relevanten Modellelemente in Choreografien
<i>FlowNode</i>	Abstrakte Oberklasse für Elemente, die als Quellen und Ziele durch Sequenzflüsse verbunden werden können
<i>ChoreographyActivity</i>	Abstrakte Oberklasse, die eine Interaktion zwischen einen oder mehreren Teilnehmern repräsentiert (Choreografie-Aktivität)
<b>ChoreographyTask</b>	Atomare Interaktion zwischen zwei Teilnehmern, die ein oder zwei Nachrichtenaustausche umfasst
<b>SubChoreography</b>	Zusammengesetzte Interaktion zweier oder mehrerer Teilnehmer, die wiederum als Repräsentation des Konzepts Choreografie gilt
<b>CallChoreography</b>	Platzhalter für die Einbindung einer weiteren Choreografie in die aktuelle Choreografie
<i>ChoreographyLoopType</i>	Repräsentation des Wiederholungsverhaltens einer Choreografie-Aktivität
<b>Message</b>	Gegenstand des Nachrichtenaustauschs zwischen zwei Teilnehmern
<b>MessageFlow</b>	Repräsentation von Nachrichtenaustausch zwischen zwei Teilnehmern
<b>SequenceFlow</b>	Repräsentation der Ausführungsreihenfolge von Choreografie-Aktivitäten
<i>Expression</i>	Repräsentation einer unterspezifizierten Bedingung in natürlicher Sprache, die für einen Sequenzfluss gilt



<i>Gateway</i>	Abstrakte Oberklasse zur Repräsentation von konvergierenden und divergierenden Verzweigungen in Sequenzflüssen
<b>ExclusiveGateway</b>	Darstellung alternativer Ausführungsreihenfolgen innerhalb einer Choreografie, wobei ausschließlich eine Alternative zur Ausführung kommt
<b>InclusiveGateway</b>	Darstellung alternativer Ausführungsreihenfolgen innerhalb einer Choreografie, wobei mehrere Alternativen zur Ausführung kommen können
<b>ParallelGateway</b>	Darstellung nebenläufiger Ausführungsreihenfolgen innerhalb einer Choreografie, die parallel zur Ausführung kommen
<i>GatewayDirection</i>	Spezifikation, ob Gateways mehrere eingehende, aber nur einen ausgehenden Sequenzfluss haben (converging), oder ob sie nur einen eingehenden, aber mehrere ausgehende Sequenzflüsse haben (diverging)

**Tabelle 2: Übersicht der identifizierten Elemente aus dem BPMN 2.0 Metamodell**

In der BPMN 2.0 können durch Events und Event-based Gateways Entscheidungen als Reaktionen auf eintreffende Ereignisse modelliert werden. Diese Entscheidungen sind jedoch passiver Natur, da die eigentliche Entscheidung von außen getroffen worden ist. Dies kann z. B. durch einen anderen Teilnehmer in der Choreografie geschehen sein, der auf Basis ihm vorliegender Informationen die eigentliche Entscheidung (z. B. per Exclusive Gateway) getroffen hat und das passive Nachvollziehen der Entscheidung bei den anderen Teilnehmern durch Nachrichtenereignisse veranlasst. Da in der DSL die aktiven Handlungen der Teilnehmer im Vordergrund stehen sollen, wurden Ereignisse und ereignisbasierte Entscheidungen nicht berücksichtigt. Aus Gründen der Verständlichkeit wurden komplexe Entscheidungen ebenso nicht berücksichtigt. Diese werden in der BPMN 2.0 als Complex Gateways modelliert und stellen Synchronisationspunkte dar, die erst bei der Ausführung einer bestimmten Anzahl eingehender Sequenzflüsse die weitere Ausführung anstoßen. Bild 3 fasst die grafische Notation der identifizierten BPMN 2.0-Untermenge zusammen. Zur Implementierung der DSL wurde das Xtext Framework aus dem Eclipse-Umfeld verwendet.



**Bild 3: BPMN 2.0 Subset für Choreografien**

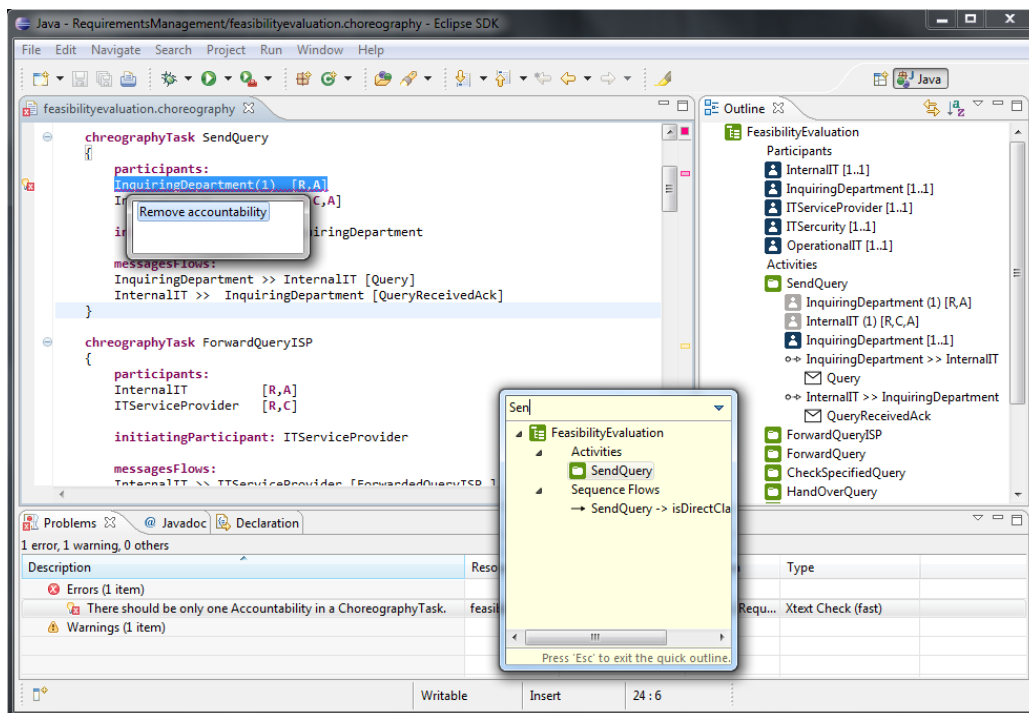


```

choreography FeasibilityEvaluation {
  participants:
    InternalIT [1..1] //Teilnehmer mit Kardinalitätsintervall
    InquiringDepartment [1..1]
    ...
  activities:
    chreographyTask SendQuery {
      participants:
        InquiringDepartment(1) [R,A] //Beteiligungen m. Anzahl u. Zuständigkeit
        InternalIT(1) [R,C]
      initiatingParticipant:
        InquiringDepartment
      messagesFlows:
        InquiringDepartment >> InternalIT [Query] //N.-fluss u. Nachricht
        InternalIT >> InquiringDepartment [QueryReceivedAck]
    }
    ...
  gateways:
    xor isDirectClassifiableSplit split //Entscheidung mit Direction
    ...
  sequenceFlows:
    SendQuery -> isDirectClassifiableSplit
    ...
}

```

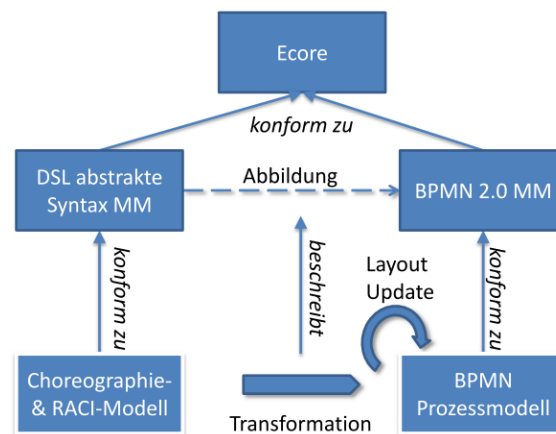
Wie am Beispiel zu sehen ist, hat die textuelle Notation den Vorteil, dass im Gegensatz zu einem grafischen Choreografiemodell das textuelle Modell gut für die iterative kollaborative Arbeit geeignet ist (Anforderung **A3**). Der Grund hierfür ist dessen blockbasierte Struktur, die eine Aufteilung des Modells auf mehrere Dokumente, sowie eine Aufteilung der Arbeit in einem einzelnen Dokument auf verschiedene Modellblöcke ermöglicht. Zudem genießen textuelle Modelle generell eine gute Werkzeugunterstützung (Anforderung **A4**). Dies betrifft besonders die notwendige Versionsverwaltung und Archivierung, wenn Choreografien als *vertragliche Werke* betrachtet werden.



**Bild 5:** DSL-Werkzeugumgebung für Choreography First-Szenarien

Bild 5 zeigt einen Ausschnitt des in Abschnitt 2 vorgestellten Szenarios als Modellierung in der mit Xtext entwickelten DSL-Werkzeugumgebung. Zur Erhöhung der Benutzerakzeptanz wurde besonderer Wert auf die fachliche Validierung und passende IDE-unterstützte Quickfixes gelegt (siehe fachliche Fehlermeldung unten links und Quickfix-Popup oben links, Anforderung **A5**). Zusätzliche Unterstützung bietet die mit BPMN-Symbolen versehene fachliche Outline View (rechts) und eine dynamische durchsuchbare Quick Outline (unten Mitte, Anforderung **A4**).

Die Interoperabilität zwischen der DSL-Werkzeugumgebung und BPMN 2.0-konformen Drittwerkzeugen wurde mittels einer *Query/View/Transformation* (QVT) Transformation hergestellt (Anforderung **A2**). QVT ist ein Standard der *Object Management Group* für bidirektionale Modell-zu-Modell-Transformationen [6]. Zur Ausführung wurde die QVT Implementierung *medini* QVT gewählt, da diese in die Eclipse-Umgebung eingebettet ist und über EMF Ecore das BPMN 2.0 Metamodell und die ebenfalls auf Ecore basierende abstrakte Syntax der DSL unterstützt (siehe Bild 6). Damit werden aus Nutzersicht Medienbrüche vermieden. Die Regeln der Transformation bilden die BPMN-Untermenge der DSL auf das vollständige BPMN 2.0-Metamodell ab. Die RACI-Zuständigkeiten werden auf BPMN 2.0 Annotationen abgebildet, um diese in BPMN 2.0-Werkzeugen sichtbar zu machen. Auf diese Weise können mit DeCREE modellierte Choreographien in beliebigen, BPMN 2.0-konformen Werkzeugen geöffnet und weiterentwickelt werden. Die Editoren bleiben unverändert und müssen nicht für die Spezialisierungen der DSL angepasst werden.



**Bild 6:** Interoperabilität durch QVT-Transformation

Zur grafischen Darstellung erfordern BPMN-Modelle Layout-Informationen. Diese werden nach Ausführung der Transformation hinzugefügt. Da die Erstellung von Layouting-Algorithmen für komplexe grafische Sprachen wie BPMN äußerst aufwendig ist wurden hier nur sehr einfache grafische Informationen hinterlegt um die Elemente darstellbar zu machen. Ein ansprechendes Layout kann in dem jeweiligen BPMN-Werkzeug vorgenommen werden.

Die QVT Transformation berücksichtigt beim erneuten Ausführen einer Transformation die Tracing-Informationen aus vorhergegangenen Ausführungen. Dadurch werden Transformationsregeln nur für solche Elemente eines Modells ausgeführt, die sich verändert haben oder hinzugekommen sind. Dies ermöglicht ein robustes Update-Verhalten. Änderungen der DeCREE-Modelle können so auch in BPMN 2.0-Modelle übertragen werden nachdem die BPMN-Modelle angepasst wurden (siehe Anforderung A3).

## 5 Zusammenfassung

In diesem Beitrag wurde anhand eines Szenarios aus der industriellen Praxis der Ansatz *Choreography-First* motiviert und eingeführt. Die ausdrückliche Modellierung der Zuständigkeiten der einzelnen Teilnehmer einer Choreografie ist essentieller Bestandteil dieses Ansatzes. Zur Unterstützung wurde die Modellierungssprache *DeCREE* entworfen und umgesetzt. Die Sprache und ihre Werkzeugunterstützung sind auf die kollaborative Modellierung durch Fachexperten zugeschnitten und konform zur BPMN 2.0. Sie sind zudem interoperabel mit BPMN 2.0-Werkzeugen, können DeCREE-Choreografiemodelle fachlich validieren und unterstützen das iterative Vorgehen in Entwicklungsprojekten. Es ist vorgesehen, sowohl Ansatz als auch Werkzeugunterstützung über das dargestellte Szenario hinaus in weiteren Projekten im Detail zu evaluieren.

## 6 Literatur

- [1] Allweyer, T (2009): Kollaborationen, Choreographien und Konversationen in BPMN 2.0. Fachhochschule Kaiserslautern, Kaiserslautern.
- [2] Awad, A; Grosskopf, A; Meyer, A; Weske, M (2009): Enabling Resource Assignment Constraints in BPMN. BPT Technical Report 04-2009, Hasso Plattner Institut, Potsdam.
- [3] Mernik, M; Heering, J; Sloane, AM (2005): When and how to develop domain-specific languages. *ACM Computing Surveys* 37(4):316-344.
- [4] Nüttgens, M; Feld, T; Zimmermann, V (1998): Business Process Modeling with EPC and UML: Transformation or Integration. In: Schader, M; Korthaus, A (Hrsg.), *The Unified Modeling Language – Technical Aspects and Applications*, Proceedings. Heidelberg, 1998.
- [5] Object Management Group. (2011): *Business Process Model and Notation (BPMN) Version 2.0*. Object Management Group.
- [6] Object Management Group (2008): *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*. Version 1.0 formal/08-04-03. Needham, MA: Object Management Group.
- [7] Rodríguez, A; Fernández-Medina, E; Piattini, M (2006): A BPMN Extension for the Modeling of Security Requirements in Business Processes. In: *IEICE Transactions* 90-D(4): 745-752 (2007).
- [8] Schleicher, D; Leymann, F; Schumm, D; Weidmann, M (2010): Compliance Scopes: Extending the BPMN 2.0 Meta Model to Specify Compliance Requirements. In: *Proceedings of IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2010)*.
- [9] Smith, ML; Erwin, J (2005): *Role and Responsibility Charting (RACI)*. [http://www.pmforum.org/library/tips/pdf\\_files/RACI\\_R\\_Web3\\_1.pdf](http://www.pmforum.org/library/tips/pdf_files/RACI_R_Web3_1.pdf). Aufgerufen am 22.09.2011.
- [10] Suarez, GN; Freund, J; Schrepfer, M (2011): *Best Practice Guidelines for BPMN 2.0*. In: L. Fischer (Hrsg), *BPMN 2.0 Handbook*. Future Strategies Inc., Lighthouse Point FL, USA.

- [11] The Eclipse Foundation (2011): Xtext 2.0. <http://www.eclipse.org/Xtext/>. Abgerufen am 21.09.2011.
- [12] Visser, E (2007): Domain-Specific Language Engineering: A Case Study in Agile DSL Development (Mark I). Delft University of Technology, Software Engineering Research Group.
- [13] Wolter, C; Meinel, C (2010): An Approach to Capture Authorisation Requirements in Business Processes. *Requirements Engineering* 15(4): 359-373 (2010).
- [14] Wolter, C; Menzel, M; Meinel, C (2008): Modeling Security Goals in Business Processes. In *Modellierung 2008, Proceedings*.
- [15] Wolter, C; Schaad, A (2007): Modeling of Task-Based Authorization Constraints in BPMN. In: Alonso, G; Dadam, P; Rosemann, M (Hrsg.), *BPM 2007, Proceedings, LNCS* 4714.